PYTHON · ШПАРГАЛКА • Модуль `re` — полный справочник

Регулярные выражения Python / Regex

8 РАЗДЕЛОВ | 60+ ПРИМЕРОВ | A4 PDF

ЧТО — символы и классы — буквально или классом

`\d \w \s` — Unicode по умолчанию. ASCII только через `re.A`

ЗАДАЧА	СИМВОЛ	СОВПАДАЕТ С
Конкретный символ	`a` `7` `@`	буквально этот символ
Любая цифра	`\d`	0–9 и Unicode-цифры
Буква / цифра / _	`\w`	лат., кирилл., цифры, _
Пробельный символ	`\s`	пробел, \t, \n, \r, \f, \v
Обратное	`\D` `\W` `\S`	всё что НЕ в \d \w \s
Один из набора	`[аеиоу]`	любой из перечисленных
Любой кроме	`[^0-9]`	всё что не в скобках
Диапазон	`[a-z]` `[а-я]`	любой в диапазоне
Буквально любой	`.`	любой символ кроме \n
То или другое	`cat|dog`	cat или dog целиком

! `.` не ловит `\n` — добавь `re.S`
! `^` внутри `[...]` — отрицание только если первый: `[^abc]` — не a,b,c; `[a^b]` — буквально `^`

СКОЛЬКО — квантификаторы — жадный / ленивый

Стоит сразу после символа или группы и относится только к ним.

КВАНТИФ.	БЕРЁТ	РЕЖИМ
`?`	0 или 1	жадный
`+`	1 или больше	жадный
`*`	0 или больше	жадный
`{n}`	ровно n	—
`{n,m}`	от n до m	жадный
`?? +? *? {n,m}?`	то же, минимум	ленивый

СТРОКА	ПАТТЕРН	РЕЗУЛЬТАТ
`<b>текст</b>`	`<.+>`	вся строка — жадный
`<b>текст</b>`	`<.+?>`	`<b>` — ленивый

! `*` совпадает с пустой строкой — `re.findall(r'\d*', 'abc')` → `['','','','']`
! В `{2,5}` — без пробела: `{2, 5}` молча ничего не найдёт

ГДЕ — якоря и границы — позиция, не символ

ЯКОРЬ	ПОЗИЦИЯ	С ФЛАГОМ RE.M
`^`	начало строки	начало каждой строки
`$`	конец строки	конец каждой строки
`\A`	начало текста	не меняется
`\Z`	конец текста	не меняется
`\b`	граница слова	между \w и не-\w
`\B`	не граница	внутри слова

! `$` совпадает перед финальным `\n` — для жёсткой границы используй `\Z`
! `_` входит в `\w` — граница \b есть между _ и пробелом, но не между _ и буквой
! Проверка всей строки — `re.fullmatch()` надёжнее чем `^паттерн$`

ЧТО РЯДОМ — lookaround — смотрит, не захватывает

Проверяет соседей без их захвата. Позиция не сдвигается.

КОНСТРУКЦИЯ	УСЛОВИЕ	ПРИМЕР
`X(?=Y)`	после X стоит Y	`\d+(?= руб)` → число перед «руб»

ГРУППЫ — захват и извлечение — по индексу или имени

Скобки группируют и сохраняют. Только сгруппировать — используй `(?:)`

СИНТАКСИС	СОХР.	КОГДА ИСПОЛЬЗОВАТЬ
`(abc)`	✅ по индексу	нужно значение
`(?:abc)`	❌	только группировка
`(?P<name>abc)`	✅ по имени	групп много — индексы путают
`\1` / `(?P=name)`	ссылка	повторить: `(\w+)\s\1` → «да да»

findall с **одной** группой → `list[str]` содержимого группы
findall с **двумя+** → `list[tuple]`
! Нужна вся строка — убери группы или оберни всё в одну

МЕТОДЫ — найти, заменить, разбить — API модуля re

ЧТО НУЖНО	МЕТОД	ВОЗВРАЩАЕТ
Есть совпадение?	`re.search(p, s)`	Match / None
Вся строка = паттерн?	`re.fullmatch(p, s)`	Match / None
Все совпадения списком	`re.findall(p, s)`	list[str] / list[tuple]
Совпадения с позициями	`re.finditer(p, s)`	iter[Match]
Заменить текстом	`re.sub(p, repl, s)`	str
Заменить функцией	`re.sub(p, func, s)`	str — func(match) → str
Разбить строку	`re.split(p, s)`	list[str]
Многократное использ.	`re.compile(p, flags)`	Pattern

! `re.match` ищет только с начала строки — частая причина «паттерн верный, ничего не находит». Сомневаешься — `re.search`

MATCH — что внутри объекта — группы, позиции, словарь

Паттерн `r'(?P<y>\d{4})-(?P<m>\d{2})-(?P<d>\d{2})'` в строке `'2024-03-15'`

ЧТО ХОЧУ	МЕТОД	РЕЗУЛЬТАТ
Всё совпадение	`.group(0)`	`'2024-03-15'`
Группа по номеру	`.group(1)`	`'2024'`
Группа по имени	`.group('y')`	`'2024'`
Все группы кортежем	`.groups()`	`('2024', '03', '15')`
Именованные словарём	`.groupdict()`	`{'y': '2024', 'm': '03', ...}`
Позиция в строке	`.span()`	`(0, 10)`

! Всегда `if m:` перед `.group()` — если нет совпадения, вернётся `None` и будет `AttributeError`

ФЛАГИ — изменить поведение движка — комбинируй через |

Комбинируй через `|`. Встроенные `(?im)` — только в начале паттерна (Python 3.11+)

ФЛАГ	КРАТКО	ЧТО МЕНЯЕТ	ПОДВОХ
`re.IGNORECASE`	`re.I`	регистр не важен	—
`re.MULTILINE`	`re.M`	`^ $` → каждая строка	\A \Z не трогает
`re.DOTALL`	`re.S`	`.` ловит `\n`	легко захватить лишнее
`re.VERBOSE`	`re.X`	пробелы и # — комментарии	пробел → пиши `\` или `[ ]`
`re.ASCII`	`re.A`	\w \d \s только ASCII	кириллица не совпадает

ГОТОВЫЕ ПАТТЕРНЫ — СКОПИРУЙ И ИСПОЛЬЗУЙ

EMAIL
```
r'[\w.+-]+@[\w-]+\.[a-zA-Z]{2,}'
```

PHONE_RU
```
r'(\+7|8)[\s\-]?\(?\d{3}\)?[\s\-]?\d{3}[\s\-]?\d{2}[\s\-]?\d{2}'
```

URL
```
r'https?://[^\s<>"{}|\\^`\[\]]+'
```

IP_BASIC
```
r'\b(?:\d{1,3}\.){3}\d{1,3}\b'
```
формат, не валидность диапазона

DATE_ISO
```
r'\d{4}-(?:0[1-9]|1[0-2])-(?:0[1-9]|[12]\d|3[01])'
```


ОНЛАЙН-КУРС · PYTHON REGEX
Забудь всё, что знал о regex. Научим правильно.
72 урока • более 900 задач • От джуна до senior-уровня
72 УРОКА | 427 ТЕСТОВ | 511 ЗАДАЧ
HTTPS://STEPIK.ORG/A/271005
ПРОМОКОД MYREGEX20 -20%